AppleScript-Hilfe

Wichtige Themen

AppleScript

Bei der Arbeit mit Ihrem Mac werden Sie feststellen, dass sich bestimmte Aufgaben häufig wiederholen. AppleScript bietet Ihnen die Möglichkeit, Skripts und Programme zu schreiben, mit denen sich wiederholende Aufgaben automatisiert werden können.

AppleScript ist eine leistungsstarke und vielseitige Skriptsprache, die in OS X integriert ist. Sie können mithilfe von AppleScript Kurzbefehle erstellen, sich wiederholende Vorgänge automatisieren oder sogar angepasste Programme erstellen, mit denen Sie viel Zeit sparen können. Mit AppleScript können Sie skriptfähige Programme steuern.

Ein AppleScript-Skript besteht aus dem Englischen ähnlichen Befehlen, mit denen Aktionen für Objekte ausgeführt werden. Beispiele für Aktionen sind „open“ (Öffnen), „close“ (Schließen) oder „get the name of“ (Ermitteln des Namens). Beispiele für Objekte sind Programme, Fenster oder Einstellungen.

Mit einem Skript können ein oder mehr Programme gesteuert werden. Für die Skriptfähigkeit müssen in einem Programm die Syntaxelemente definiert sein, die Sie in Ihren Skripts verwenden können. Zum Lieferumfang von OS X gehören mehrere skriptfähige Programme.

Mit Skripts können Sie auch Programme in einem Netzwerk steuern, sodass Sie Aufgaben auf mehr als einem Computer automatisieren können.

Faktisch haben Sie AppleScript bereits verwendet, ohne es zu wissen. Viele Funktionen im System und in Programmen basieren auf AppleScript-Skripts, die innerhalb eines Programms aufgerufen werden. Wenn Sie zum Beispiel in diesem Hilfesystem auf einen Link klicken, der ein Programm oder eine Datei öffnet, führt ein AppleScript-Skript die Aktion aus.

VERWANDTE THEMEN

Funktionsweise von AppleScript
AppleScript-Editor
Die Sprache „AppleScript“
AppleScript-Ressourcen im Internet
Entwicklungsumgebung von AppleScript Studio

Wichtige Themen

AppleScript-Editor

Der AppleScript-Editor ist ein benutzerfreundliches Programm zum Erstellen von Skripts, in dem AppleScript-Skripts gelesen, geschrieben, aufgezeichnet und gesichert werden können.

Einige Leistungsmerkmale des AppleScript-Editors sind:

- Er stellt eine einheitliche, besonders benutzerfreundliche Oberfläche zur Verfügung.
- Er ermöglicht das Suchen und Ersetzen von Text in Skripts.
- Lange Skriptanweisungen werden automatisch umbrochen.

- Der Befehl „Widerrufen“ kann mehrmals ausgeführt werden.
- Er stellt Skripts im Kontextmenü zur schnelleren Skripterstellung zur Verfügung.
- Programmdienste werden unterstützt.
- Sie können mithilfe der Bibliothekspalette auf Programmskripts zugreifen.
- Er stellt einen Skriptassistenten zur Verfügung, der Sie beim Eingeben von Anweisungen unterstützt.
- Er stellt die Optionen „Verlauf der Ergebnisse“ und „Verlauf des Event-Protokolls“ zur Verfügung, mit denen Sie die Skriptabläufe überprüfen können.

VERWANDTE THEMEN

Die Sprache „AppleScript“
Entwicklungsumgebung von AppleScript Studio

Wichtige Themen

Erstellen von AppleScript-Skripts

Sie müssen gar nichts über Programmierung wissen, um von AppleScript profitieren zu können. Genau genommen, müssen Sie keine einzige Skriptzeile selber schreiben. In OS X sind hunderte nützlicher AppleScript-Skripts enthalten, die Sie verwenden können, und tausende sind im Web verfügbar.

Wenn Sie einen Schritt weitergehen möchten, können Sie vorhandene Skripts als Grundlage nehmen und sie mit kleineren Änderungen an Ihre Anforderungen anpassen.

Am meisten profitieren Sie natürlich, wenn Sie eigene Skripts erstellen. Wenn Sie das Schreiben von Skripts lernen, haben Sie die Möglichkeit, nützliche Tools und zeitsparende Automatisierungen zu schreiben, sodass Sie produktiver arbeiten können.

Mit dem AppleScript-Editor, der eine benutzerfreundliche Skritping-Umgebung bereitstellt, können Sie eigene Skripts erstellen und ändern. Alle Beispiele in der AppleScript-Hilfe wurden mithilfe des AppleScript-Editors erstellt.

- Öffnen Sie den AppleScript-Editor.

  AppleScript-Editor öffnen

  Geben Sie den folgenden Text in das Fenster „AppleScript-Editor“ ein.

  ```
  say "Willkommen bei AppleScript!"
  ```

  Klicken Sie auf „Ausführen“.

VERWANDTE THEMEN

AppleScript-Editor
Entwicklungsumgebung von AppleScript Studio

Wichtige Themen

Funktionsweise von AppleScript

AppleScript sendet als „Apple-Events“ bezeichnete Nachrichten. Wenn Sie ein Skript ausführen, sendet AppleScript Apple-Events an das System oder an die Programme, von denen die Befehle ausgeführt werden.

In vielen Fällen geben Programme oder das System ein Ergebnis an AppleScript zurück. Wenn z. B. ein Skript eine Anforderung an ein Programm sendet, um den Namen eines Objekts abzurufen, gibt das Programm den Namen zurück. Wenn der Befehl nicht ausgeführt werden kann, gibt das Programm oder AppleScript eine Fehlermeldung zurück. Alle Ergebnisse werden im AppleScript-Editor angezeigt.

Im Allgemeinen müssen Sie die Funktionsweise von Apple-Events nicht kennen. AppleScript führt alle nötigen Schritte im Hintergrund aus. Sie müssen Ihre Skripts nur schreiben und ausführen.

VERWANDTE THEMEN

AppleScript
Die Sprache „AppleScript“

Applets

Ein Skript, das als Programm gesichert wurde, wird als AppleScript-Programm oder -Applet bezeichnet. Applets werden beim Öffnen automatisch ausgeführt, wodurch zugleich die Befehle im Skript ausgeführt werden. Applets können nur ein paar Befehle, aber auch komplexe Programme mit Tausenden von Zeilen enthalten.

Applets können als bearbeitbar oder als nur ausführbar gesichert werden. Ein bearbeitbares Applet enthält noch den Text des Skripts, sodass es jederzeit bearbeitet und neu übersetzt werden kann. Bewegen Sie ein Applet auf das Symbol des AppleScript-Editors, um es zu bearbeiten oder zu ändern. Wenn Sie ein Skript als nur ausführbar sichern, wird der bearbeitbare Textteil gelöscht, sodass die Datei nicht mehr bearbeitet oder neu übersetzt werden kann.

Zum Bearbeiten eines Applets, das als bearbeitbar gesichert wurde, öffnen Sie es im AppleScript-Editor oder einem anderen AppleScript-Editorprogramm. Applets erfordern eine „on run“-Routine, die mit der Ausführung des Skripts beginnt, wenn das Applet geöffnet wird.

- Das Beispiel zeigt eine „on run“-Routine.

```
on run
        --erster Befehl
        --zweiter Befehl
end run
```

VERWANDTE THEMEN

Droplets
AppleScript-Dateitypen

Übersetzte Skripts

Ein übersetztes (kompiliertes) Skript ist eine AppleScript-Datei in bearbeitbarer Form, die Sie im AppleScript-Editor oder einem anderen AppleScript-Editorprogramm lesen und bearbeiten können.

Wenn Sie ein übersetztes Skript öffnen, wird es im AppleScript-Editor anzeigt, ohne dass die darin enthaltenen Befehle ausgeführt werden. Dies ermöglicht Ihnen, Änderungen an dem Skript vorzunehmen, ohne es auszuführen. Anschließend können Sie das Skript übersetzen. Dabei wird der gesamte Text in der Datei gelesen und in AppleScript-Code konvertiert. Das übersetzte Skript kann ausgeführt werden. Ein übersetztes Skript hat die Dateierweiterung „.scpt“.

Wenn Sie die Datei als übersetztes Skript sichern, bleibt der Text zusammen mit dem Code in der Datei erhalten. Sie können das übersetzte Skript deshalb jederzeit erneut bearbeiten.

VERWANDTE THEMEN

Applets
Droplets
AppleScript-Dateitypen

Droplets

Ein Droplet ist ein AppleScript-Skript, das ausgeführt wird, wenn Sie Dateien oder Ordner auf das zugehörige Symbol bewegen. Das Skript führt Aktionen für die Dateien oder Ordner aus.

Wie Applets können auch Droplets als bearbeitbar oder als nur ausführbar gesichert werden. Ein bearbeitbares Droplet enthält noch den Text des Skripts, sodass es jederzeit bearbeitet und neu übersetzt werden kann. Bewegen Sie ein Droplet auf das Symbol des Skript-Editors, um es zu bearbeiten oder zu ändern. Wenn Sie ein Skript als nur ausführbar sichern, wird der bearbeitbare Textteil gelöscht, sodass die Datei nicht mehr bearbeitet oder neu übersetzt werden kann.

Droplets erfordern eine „on open“-Routine, die mit der Ausführung des Skripts beginnt, wenn das Droplet geöffnet wird. Die „on open“-Routine weist einen Parameter auf, der eine Liste der auf das Droplet bewegten Objekte enthält. In diesem Teil des Droplet-Skripts wird die Liste untersucht, und in einem anderen Teil werden die Objekte verarbeitet. Sofern durch das Droplet nicht nur ein einzelnes Objekt verarbeitet wird, sollte es eine „repeat“-Schleife beinhalten, in der die Objekte in der Liste durchlaufen werden, wobei für jedes Objekt nacheinander Aktionen ausgeführt werden.

- Beispiel für eine Droplet-Routine, mit der „on open“- und der Parameterliste.

```
on open (theItemsDropped)
        --erster Befehl
        --zweiter Befehl
end open
```

VERWANDTE THEMEN

Applets
Übersetzte Skripts
AppleScript-Dateitypen

Verwenden von AppleScript-Funktionsverzeichnissen

Sehen Sie sich das Funktionsverzeichnis eines Programms an, um herauszufinden, welche Befehle und Objekte Sie zum Steuern des betreffenden Programms verwenden können. Ein Funktionsverzeichnis enthält eine Gruppe von Begriffen, die

ein Programm verarbeiten kann.

Sie können das Funktionsverzeichnis eines Programms im AppleScript-Editor anzeigen, indem Sie das Programm im Fenster „Bibliothek" auswählen oder das Symbol des Programms auf das Symbol des AppleScript-Editors bewegen.

Die Begriffe in einem Funktionsverzeichnis sind zu Gruppen zusammengefasst, die als „Suite" bezeichnet werden. Befehle und Objekte sind fettgedruckt.

---

1. Wählen Sie im AppleScript-Editor „Fenster" > „Bibliothek".
2. Öffnen Sie den Namen des Programms durch Doppelklicken im Fenster „Bibliothek".
3. Wählen Sie eine Suite (Gruppe) aus.

   Die Begriffe in dieser Suite und die zugehörigen Beschreibungen werden angezeigt.

Die Sprache AppleScript umfasst verschiedene Kategorien:

- Suites: Gruppen zusammengehöriger Begriffe. Im Funktionsverzeichnis des AppleScript-Editors werden Suites in der ersten Spalte angezeigt und sind mit einem orangefarbenen Buchstaben „S" gekennzeichnet.
- Befehle: Befehle fordern eine Aktion oder ein Ergebnis an. Befehle, die auch als „Events" bezeichnet werden, werden in der zweiten Spalte angezeigt und sind mit einem blauen Buchstaben „C" gekennzeichnet.
- Klassen: skriptfähige Objekte. Klassenbeschreibungen geben die in der Klasse vorhandenen Eigenschaften und Elemente an. Klassen werden in der zweiten Spalte angezeigt und sind mit einem lavendelfarbenen Buchstaben „C" gekennzeichnet.
- Eigenschaften: Objekte, die Werte enthalten. Eigenschaften werden in der dritten Spalte angezeigt und sind mit einem lavendelfarbenen Buchstaben „P" gekennzeichnet.
- Elemente: Objekte, die in anderen Objekten enthalten sind. Zum Beispiel kann ein Absatz viele Wörter enthalten. Elemente werden beispielsweise in der dritten Spalte angezeigt und sind mit einem orangefarbenen Buchstaben „e" gekennzeichnet.

VERWANDTE THEMEN

Klassendefinitionen
Befehlsdefinitionen
Verwenden von AppleScript zum Steuern von Programmen

FOLGENDES ÖFFNEN

AppleScript Language Guide: Commands Reference
AppleScript Language Guide: Class Reference

Wichtige Themen

Fehlerbeseitigung

Sollten bei der Verwendung von AppleScript Probleme auftreten, finden Sie in den folgenden Themen nützliche Hinweise zur Fehlerbeseitigung.

Ein Fehler wird angezeigt, wenn ein Benutzer in einem Dialogfenster „choose from list" auf „Abbrechen" klickt

Ihre Ordneraktion wird unerwartet ausgelöst

Klassendefinitionen

In einer Klassendefinition werden Eigenschaften und Elemente eines skriptfähigen Objekts beschrieben. Sie können eine Klasse mit einem Substantiv in einem Satz vergleichen, und eine Klassendefinition wäre dementsprechend die Definition des Substantivs in einem Wörterbuch.

Gewöhnlich beginnt eine Klassendefinition in einem AppleScript-Funktionsverzeichnis mit dem fett gedruckten Begriff und einer kurzen Beschreibung des Begriffs in derselben Zeile.

Dann folgt eine Liste aller Elemente oder Objekte, die zu dem skriptfähigen Objekt gehören oder aus denen dieses sich zusammensetzt. Die Klassendefinitionen enthalten nicht grundsätzlich eine Liste der Elemente, da einige skriptfähige Objekte andere skriptfähige Objekte enthalten und einige nicht.

Die Klassendefinition wird mit einer Liste der Eigenschaften (Properties) oder Qualitäten des betreffenden skriptfähigen Objekts sowie den entsprechenden Werten abgeschlossen.

Ein Eigenschaftseintrag beginnt mit dem Namen der Eigenschaft und wird ergänzt durch:

- Eine Beschreibung des Datenformats zur Speicherung des Werts der Eigenschaft, beispielsweise „integer" (Ganzzahl), „point" (eine Liste mit zwei Zahlen), „string" (Textdaten), „Boolean" (zutreffend (true) oder unzutreffend (false)) etc.
- Einen optionalen Typindikator, [r/o], durch den angegeben wird, dass sich der Wert der Eigenschaft im Status „Nur Lesen" befindet und nur gelesen und nicht geändert werden kann.
- Eine kurze Beschreibung der Eigenschaft.

Gibt es für das Objekt eine Pluralform, können Sie auf mehr als ein Objekt verweisen.

Ist für ein Objekt eine Liste von Elementen vorhanden, enthält es weitere Objekte. Der Name des Elements, den Sie in Ihrem Skript verwenden können, erscheint in Fettdruck. Der Text in Normalschrift beschreibt, wie auf das Element verwiesen werden muss.

Objekte können über Eigenschaften verfügen. Die Namen von Eigenschaften werden in Fettdruck angezeigt. Der Text in Normalschrift beschreibt den Wertetyp, den die Eigenschaft enthält, etwa ein Text. Eigenschaften können ein konstanter Wert oder ein Wert aus einer Liste von Konstanten sein. Ein Skript kann den Wert einer Eigenschaft erhalten. Es kann den Wert einer Eigenschaft festlegen, sofern nach der Syntax nicht der Vermerk „[r/o]" (Nur Lesen) steht.

VERWANDTE THEMEN

Befehlsdefinitionen
Objektdefinitionen
AppleScript-Befehle
Verwenden von AppleScript-Funktionsverzeichnissen

FOLGENDES ÖFFNEN

AppleScript Language Guide: Class Reference

Befehlsdefinitionen

Befehle sind Begriffe in AppleScript, mit denen Aktionen ausgeführt werden. Befehle in einem AppleScript-Funktionsverzeichnis werden in der mittleren Spalte angezeigt und sind mit einem blauen Buchstaben „C“ gekennzeichnet. Die Definition eines Befehls beinhaltet eine kurze Beschreibung dessen, was er bewirkt, und seine Syntax. Die Syntax legt fest, wie Sie den Befehl in einem Skript eingeben müssen. Die Syntax kann aus verschiedenen Teilen bestehen.

Wörter in Fettdruck sind die Befehle, die Sie in einer Anweisung verwenden. Sie müssen diese Befehle exakt so schreiben, wie sie im Funktionsverzeichnis stehen.

Blau geschriebene Wörter kennzeichnen Informationen, die Sie angeben. So können Sie z. B. ein Objekt oder eine von Ihnen definierte Variable angeben.

Befehle enthalten manchmal optionale Teile, die in Klammern stehen. Sie müssen diese nicht in einer Anweisung verwenden.

VERWANDTE THEMEN

Klassendefinitionen
Verwenden von AppleScript-Funktionsverzeichnissen
Event-Definitionen
Objektdefinitionen

AppleScript Language Guide: Commands Reference

Event-Definitionen

In einer Event-Definition wird ein Verb oder Befehl zum Anwenden einer Aktion auf ein skriptfähiges Objekts beschrieben. Events sind Aktionen oder Verben, die von Objekten ausgeführt werden können oder an denen sich Objekte beteiligen können.

Gewöhnlich beginnt eine Event-Definition mit dem jeweiligen Befehl und einer kurzen Beschreibung des Befehls in derselben Zeile.

Dann folgt eine Liste aller Parameter, die zur Ausführung des Befehls verwendet werden. Optionale Parameter werden in Klammern dargestellt, erforderliche Parameter nicht.

Jeder Parametereintrag beginnt mit dem Namen des Parameters, gefolgt von einer Beschreibung des mit dem Parameter verwendeten Datenformats. Der Eintrag endet mit einer kurzen Beschreibung des Parameters selbst.

Die Definition eines Befehls enthält eine kurze Beschreibung (der unterstrichene Text) der Funktionsweise und Syntax des Befehls. Die Syntax legt fest, wie Sie den Befehl in einem Skript schreiben. Die Syntax kann aus verschiedenen Teilen bestehen.

Wörter in Fettdruck sind die Befehle, die Sie in einer Anweisung verwenden. Sie müssen diese Befehle exakt so schreiben, wie sie im Funktionsverzeichnis stehen.

Wörter in Normalschrift sind die von Ihnen bereitgestellten Informationen. So können Sie z. B. ein Objekt oder eine von Ihnen definierte Variable angeben.

Befehle enthalten manchmal optionale Teile, die in Klammern stehen. Sie müssen diese nicht in einer Anweisung verwenden.

Befehle manipulieren Objekte. Eine Objektdefinition enthält die Syntax für die Beschreibung des Objekts, einschließlich des Namens des Objekts und einer kurzen Beschreibung (der unterstrichene Text). Die Klasse eines Objekts bestimmt spezielle Objektmerkmale, etwa die Befehle, die Sie zur Steuerung des Objekts verwenden können.

VERWANDTE THEMEN

Database Events
Klassendefinitionen
Befehlsdefinitionen
Objektdefinitionen
Funktionsweise von AppleScript

Objektdefinitionen

Befehle manipulieren Objekte. Eine Objektdefinition in einem AppleScript-Verzeichnis enthält die Syntax für die Beschreibung des Objekts, einschließlich des Namens des Objekts und einer kurzen Beschreibung (der unterstrichene Text). Die Klasse eines Objekts bestimmt spezielle Objektmerkmale, etwa die Befehle, die Sie zur Steuerung des Objekts verwenden können.

Gibt es für das Objekt eine Pluralform, können Sie auf mehr als ein Objekt verweisen.

Ist für ein Objekt eine Liste von Elementen vorhanden, enthält es weitere Objekte. Der Name des Elements, den Sie in Ihrem Skript verwenden können, erscheint in Fettdruck. Der Text in Normalschrift beschreibt, wie auf das Element verwiesen werden muss.

Objekte können über Eigenschaften verfügen. Eigenschaftsnamen sind blau und fett gedruckt. Der blaue Text in Normalschrift beschreibt den Wertetyp, den die Eigenschaft enthält, etwa ein Text. Eigenschaften können ein konstanter Wert oder ein Wert aus einer Liste von Konstanten sein. Ein Skript kann den Wert einer Eigenschaft erhalten. Es kann den Wert einer Eigenschaft festlegen, sofern nach der Syntax nicht der Vermerk „[r/o]“ (Nur Lesen) steht.

VERWANDTE THEMEN

Verwenden von AppleScript zum Steuern von Programmen
Befehlsdefinitionen
Klassendefinitionen
tell-Anweisungen für Objekte
Verwenden von AppleScript-Funktionsverzeichnissen

FOLGENDES ÖFFNEN

AppleScript Language Guide: Class Reference
AppleScript Language Guide: Properties

---

if-Anweisungen

Eine if-Anweisung ist ein Befehl, der nur dann ausgeführt wird, wenn bestimmte Bedingungen erfüllt sind. Jede if-Anweisung enthält einen booleschen Ausdruck, der entweder zutreffend (true) oder unzutreffend (false) ist. Die Befehle innerhalb der if-Anweisung werden nur ausgeführt, wenn der Wert des booleschen Ausdrucks zutreffend (true) ist. Die if-Anweisung endet mit „end if“. Hier ein Beispiel:

```
if amountReceived > 100 then
        set memberStatus to "special"
end if
```

In der obigen Anweisung werden die Befehle innerhalb der if-Anweisung ausgeführt, wenn der boolesche Ausdruck zutreffend (true) ist (wenn der erhaltene Betrag (amountReceived) größer als 100 ist). Wenn der boolesche Ausdruck unzutreffend (false) ist, werden die Befehle nicht ausgeführt.

Die if-Anweisung kann auch mit einer else-Anweisung verknüpft werden. Dieses Konstrukt wird als „if-then-else"-Anweisung bezeichnet. Hier ein Beispiel:

```
if amountReceived > 100 then
        set memberStatus to "special"
else
        set memberStatus to "general"
end if
```

In der obigen Anweisung werden die Befehle im ersten Teil der Anweisung ausgeführt, wenn der erste boolesche Ausdruck (amountReceived größer als 100) zutreffend (true) ist. Wenn der erste boolesche Ausdruck unzutreffend (false) ist, werden die Befehle im zweiten Teil der Anweisung nach „else" ausgeführt.

Sie können if-Anweisungen ineinander verschachteln, um mehrere Bedingungen zu verarbeiten.

Weitere Informationen über if-Anweisungen finden Sie im AppleScript Language Guide.

VERWANDTE THEMEN

repeat-Anweisungen
Befehlsdefinitionen
Definieren von Variablen in AppleScript
Konstante Werte

FOLGENDES ÖFFNEN

AppleScript Language Guide - If

Einfache Befehle

repeat-Anweisungen

Sie können repeat-Anweisungen verwenden, um je nach Bedingung Aufgaben mehrmals durchzuführen. Durch eine repeat-Anweisung wird eine Gruppe von Anweisungen so lange in einer Schleife ausgeführt, bis eine Bedingung erfüllt ist und die Schleife verlassen wird. Sie können repeat-Anweisungen ineinander verschachteln, um auf mehrere Bedingungen zu reagieren.

AppleScript bietet eine Palette von repeat-Anweisungen, die auf unterschiedliche Bedingungen reagieren können.

---

repeat (forever)

Kontinuierliches Wiederholen in einer Schleife. Verwenden Sie zum Verlassen der Endlosschleife eine exit-Anweisung.

```
repeat
        -- Anweisung
```

```
        -- Anweisung
        if (x = 5) then exit repeat
end repeat
```

---

repeat (Zahl) times

So oft wiederholen, wie angegeben.

```
repeat 5 times
        -- Anweisung
        -- Anweisung
end repeat
```

---

repeat while

Wiederholen, solange eine angegebene Bedingung zutreffend (true) ist.

```
set x to 1
repeat while (x < 5)
        -- Anweisung
        -- Anweisung
        set x to (x + 1)
end repeat
```

---

repeat until

Wiederholen, bis eine angegebene Bedingung zutreffend (true) ist.

```
set x to 1
repeat until (x = 5)
        -- Anweisung
        -- Anweisung
        set x to (x + 1)
end repeat
```

---

repeat with (Variable) from (Startwert) to (Stoppwert)

Setzt die Variable auf den „Startwert“ und erhöht dann den Variablenwert nach jedem Schleifendurchlauf. Die Wiederholung endet, wenn die Variable den „Stoppwert“ erreicht.

```
repeat with x from (5) to (25)
        -- Anweisung
        -- Anweisung
end repeat
```

---

repeat with (Variable) in (Liste)

Setzt die Variable auf das erste Objekt in der Liste. Bei jedem Durchlauf der Schleife wird die Variable auf das nächste Objekt gesetzt. Die Wiederholung endet, nachdem alle Objekte von „Liste“ verwendet wurden.

```
set meine_Liste to {234, 342, 681, 874}
repeat with x in meine_Liste
        -- Anweisung
        -- Anweisung
end repeat
```

Weitere Informationen über repeat-Anweisungen finden Sie im AppleScript-Handbuch.

VERWANDTE THEMEN

if-Anweisungen
Definieren von Variablen in AppleScript
Konstante Werte

FOLGENDES ÖFFNEN

AppleScript Language Guide - Repeat

Konstante Werte

Eine Konstante ist ein reserviertes Wort in AppleScript. Sie enthält einen vorgegebenen Wert. AppleScript enthält zahlreiche Konstanten, die Sie verwenden können, um Informationen über Objekte zu erhalten, Vergleiche anzustellen oder arithmetische Berechnungen durchzuführen. Programme und andere Objekte können eigene Konstanten definieren. Diese werden in den Wertklassendefinitionen definiert.

Wenn Sie den Wert einer Konstante sehen möchten, setzen Sie das folgende Beispiel im AppleScript-Editor ein und klicken Sie dann auf „Ausführen". Klicken Sie dann auf „Ergebnis", um den Wert der Konstante anzuzeigen.

Das Ergebnis ist ebenfalls eine Konstante. Es handelt sich dabei um eine Skriptkonstante, die sich auf den von einem Befehl zurückgegebenen Wert bezieht.

```
get the current date
```

Einige weitere Beispiele für Konstanten:

- Arithmetische Konstanten: pi
- Boolesche Konstanten: true, false
- Datums- und Zeitkonstanten: minutes, hours, days, weeks
- Textkonstanten: return, space, tab
- Skriptkonstanten: anything, current application, it, me, missing value, my, result

Weitere Informationen über Konstanten in AppleScript finden Sie im AppleScript Language Guide.

VERWANDTE THEMEN

Definieren von Variablen in AppleScript
Objektdefinitionen
Klassendefinitionen
Event-Definitionen

FOLGENDES ÖFFNEN

AppleScript Language Guide - Constants

Einfache Befehle

Definieren von Variablen in AppleScript

Eine Variable ist ein innerhalb eines Skripts erstelltes Container-Objekt, in dem ein Wert gespeichert wird. Sie geben beim Erstellen einen Namen bzw. Bezeichner für die Variable an. Wenn ein Skript ausgeführt wird, können Sie der Variablen einen Wert zuweisen, auf den Wert zugreifen oder ihn ändern. Variablen bleiben nur für die Dauer der Skriptausführung erhalten. Wenn das Skript geschlossen wird, geht der Wert verloren.

- Diese Beispiele zeigen, wie Sie einen eindeutigen Bezeichner angeben und diesem einen Wert zuordnen.

```
set meine_Variable to "153"

set x to "Quartalsberichte"

set x to false
```

Wenn Sie sehen möchten, wie Werte in einer Variablen gespeichert werden, setzen Sie diese Beispiele im AppleScript-Editor ein und klicken Sie dann auf „Ausführen“. Klicken Sie auf „Ergebnis“, um den in der Variablen gespeicherten Wert anzuzeigen.

Weitere Informationen über Variablen in AppleScript finden Sie im AppleScript Language Guide.

VERWANDTE THEMEN

Konstante Werte
if-Anweisungen
repeat-Anweisungen

FOLGENDES ÖFFNEN

AppleScript Language Guide - Variables

Einfache Befehle

Angeben des Programmspeicherorts

Die Programme in OS X sind in sogenannten „Paketen“ zusammengefasst. Ein Paket ist ein Ordner, der die Ordner und Dateien enthält, aus denen das Programm besteht. Aufgrund dessen können die Befehle „path to“ und „info for“ unterschiedliche Ergebnisse für Programme zurückgeben.

Wenn Sie den Befehl „path to“ für ein Programm verwenden, endet der Pfad gewöhnlich mit „.app“. Im Finder werden Dateierweiterungen u. U. nicht als Bestandteil des Dateinamens angezeigt, dennoch können Sie über Erweiterungen Dateien in einem Skript angeben.

Wenn Sie mit dem Befehl „info for“ Informationen über ein Programm abrufen, ist die Eigenschaft „Folder“ des Ergebnisses gewöhnlich zutreffend (true). Weitere Informationen zur Datei und ihrem Namen können Sie weiteren Eigenschaften des Ergebnisses wie „extension hidden“, „name extension“ oder „displayed name“ entnehmen.

VERWANDTE THEMEN

path to
info for
Verweisen auf Ordner, Dateien und Volumes
choose application
Finder-Skriptbefehle

Verweisen auf Ordner, Dateien und Volumes

AppleScript ermöglicht Ihnen, bestimmte Ordner, Dateien und Volumes mit Namen anzusprechen, ohne einen Pfad angeben zu müssen. AppleScript kann den Speicherort dieser Objekte ermitteln, etwa den des Ordners „Dokumente", des Schreibtisches, des Benutzerordners und des Startvolumes.

- Das Beispiel zeigt, wie auf bestimmte Ordner unter Verwendung ihres Namens in einem Skript verwiesen werden kann.

```
set x to the path to the desktop folder
```

VERWANDTE THEMEN

Definieren von Verweisen auf den Finder in Skripts
path to
info for
Finder-Skriptbefehle

Definieren von Verweisen auf den Finder in Skripts

Damit Sie Finder-Befehle verwenden können, übermitteln Sie sie mit einer tell-Anweisung an den Finder (ähnlich wie Befehle in einem Programm).

- Das folgende Beispiel zeigt die Verwendung einer tell-Anweisung in einem Skript. Mit einer tell-Anweisung übermitteln Sie die Funktion, die die Anzahl der Fenster ermittelt, an „Finder.app".

```
tell application "Finder.app"
        return the number of windows
end tell
```

VERWANDTE THEMEN

Finder
tell-Anweisungen für Objekte
Verweisen auf Ordner, Dateien und Volumes
Objektdefinitionen
Finder-Skriptbefehle

Finder-Befehle

Finder-Skriptbefehle

Hier sind einige Befehle, die Sie im Finder verwenden können.

activate

alias file

alias list

application file

clean up

close

column

Computer-object

container

copy

count

data size

delete

desktop window

desktop

disk

document file

duplicate

eject

empty trash

exists

file

file information

find

Finder

Finder window

folder

get

icon view options

information window

Internet location file

item

list view options

make

move

moving folder window for

open

package

path to

preferences

preferences window

print

quit

Verweisen auf Ordner, Dateien und Volumes

Definieren von Verweisen auf den Finder in Skripts

reveal

select

set

sort

trash

update

window

Finder-Befehle

Standardbefehle für Skriptzusätze

Die Skriptzusätze bieten Ihnen viele nützliche Befehle, die Sie in Ihre Skripts einfügen können.

adding folder items to

beep

choose application

choose color

choose file

choose file name

choose folder

choose from list

choose URL

clipboard info

close access

closing folder window for

current date

delay

dialog reply

display

display dialog

do shell script

download

FTP item

get eof

info for

list disks

list folder

load script

mount volume

offset

open for access

open location

opening folder

path to

POSIX path

random number

read

round

run script

say

scripting components

set eof

set the clipboard to

set volume

store script

summarize

system attribute

time to GMT

upload

URL

write